IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.: 10/679,745
Filed: October 6, 2003
Inventor(s):
Alsup et al.

Title: APPARATUS AND METHOD FOR PORT ARBITRATION IN A REGISTER FILE

Examiner: Johnson, Brian P.
Group/Art Unit: 2183
Atty. Dkt. No: 5500-89200
Confirm. No. 2802

****CERTIFICATE OF E-FILING TRANSMISSION****
I hereby certify that this correspondence is being transmitted via electronic filing to the United States Patent and Trademark Office on the date shown below

Stephen J. Curran
Printed Name

/ Stephen J. Curran /
Signature

May 16, 2007
Date

SUPPLEMENTAL APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Non-Compliant Appeal Brief mailed on April 17, 2007, Appellant presents this Supplemental Appeal Brief. As shown below, Appellant has amended Section V Summary Of Claimed Subject Matter to include more explicit references to the Appellant's specification and figures.

I. REAL PARTY IN INTEREST

The present application is owned by Advanced Micro Devices, Inc. An assignment of the present application to the owner is recorded at Reel 014587, Frame 0495.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences known to Appellant.

III. STATUS OF CLAIMS

Claims 1, 9-12, and 20-37 are pending. Claims 2-8 and 13-19 are cancelled. Claims 1, 9, 12, 20, and 23 are rejected under 35 U.S.C. § 103(a). It is these rejections that are being appealed. A copy of claims 1, 9-12, and 20-37 is included in the Claims Appendix attached hereto.

IV. STATUS OF AMENDMENTS

No unentered amendment to the claims has been filed after final rejection.

However, Appellant notes in the Advisory action dated November 21, 2006, the Examiner checked box #10 to indicate an affidavit or other evidence is entered. Appellant is unaware of any affidavit or other evidence entered, and is thus confused by this box being checked.

V. SUMMARY OF CLAIMED SUBJECT MATTER

As the number of functional units in a microprocessor implementation increases, the number of register file read and write ports that supply operands may necessarily increase. However, increasing the number of read and write ports greatly increases the complexity of designing and implementing the register file. Each additional read or write port requires additional control logic to decode and select which location to access via that port. Further, each such additional port requires a separate wire bus connected to each register file location, if access symmetry (i.e., any location being readable or writeable by any port) is to be maintained. Depending on the manufacturing process used to manufacture the microprocessor, additional wiring may require a larger physical register file layout, resulting in slower register file performance due to electrical factors such as increased capacitive loading and signal transit time. (*See* e.g., specification page 2, lines 12-20)

The performance costs of adding read or write ports may ultimately outweigh the performance benefits of increasing parallelism through adding functional units. Not providing sufficient read and write ports to simultaneously supply all functional units also limits performance. Thus, it may be desirable to have a processor design including an optimized number of register file read and write ports in a given implementation. (*See* e.g., specification page 2, lines 22-26)

Independent claim 1 is directed to a microprocessor (*See* e.g., Fig. 1, 100) configured to provide port arbitration in a register file. In one embodiment, the microprocessor includes a plurality of functional units configured to collectively operate on a maximum number of operands in a given execution cycle, and a register file providing a number of read ports that is insufficient to provide the maximum number of operands to the plurality of functional units in the given execution cycle. (*See*, e.g., Fig. 1; 142a-b, 144a-b; specification page 6, line 14-23; page 8, lines 2-5; page 11, lines 22-26). The microprocessor also includes an arbitration logic coupled to allocate the read ports of the register file for use by selected functional units during the given execution cycle. (*See*, e.g., Fig. 1; 115, specification page 7, lines 16-19) Each of the plurality of functional units corresponds to one of a plurality of issue slots. The arbitration logic is further configured to allocate a first portion of the read ports to a first portion of the plurality of functional units that corresponds to a first issue slot. (*See*, e.g., specification page 13 line 14-19; page 13, lines 23-25).

Independent claim 12 is directed to a method of operating a plurality of functional units when there are an insufficient number of read ports in a register file to provide operands to the functional units. More particularly, the method includes configuring the plurality of functional units to collectively operate on a maximum number of operands in a given execution cycle. (*See*, e.g., Fig. 6; specification page 6, line 14-23; page 8, lines 2-5; page 11, lines 22-26) The functional units are grouped into a plurality of distinct issue slots, and each of the issue slots includes at least two functional units of different types. (*See*, e.g., specification page 13, line 21-23) The method also includes providing a number of read ports that is insufficient to provide the maximum number of operands to the functional units in the given execution cycle. (*See*, e.g., specification page 6, line 14-23; page 8, lines 2-5; page 11, lines 22-26) The method further includes allocating portions of the read ports of the register file to functional units included in each of the issue slots for use during the given execution cycle. The portions are respectively associated with the issue slots. (*See*, e.g., Fig. 5, Fig. 6, and specification page 13 line 14-19; page 13, lines 23-25).

Independent claim 23 is directed to a system including a system memory coupled to a microprocessor that includes a plurality of functional units configured to collectively operate on a maximum number of operands in a given execution cycle, and a register file providing a number of read ports that is insufficient to provide the maximum number of operands to the plurality of functional units in the given execution cycle. (*See*, e.g., Fig. 1, 100; and specification page 6, line 14-23; page 8, lines 2-5; page 11, lines 22-26) The microprocessor also includes an arbitration logic coupled to allocate the read ports of the register file for use by selected functional units during the given execution cycle. (*See*, e.g., Fig. 1; 115, specification page 7, lines 16-19) Each of the plurality of functional units corresponds to one of a plurality of issue slots. The arbitration logic is further configured to allocate a first portion of the read ports to a first portion of the plurality of functional units that corresponds to a first issue slot. (*See*, e.g., specification page 13 line 14-19; page 13, lines 23-25).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1, 9, 12, 20, and 23 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Johnson (U.S. Patent No. 5,129,067) (hereinafter "Johnson") in view of Black et al. (U.S. Publication No. 2004/0064679) (hereinafter "Black").

2. Claims 10-11 and 21-22 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Johnson in view of prior art.

3. Claims 24-37 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Johnson in view of Black.

VII. ARGUMENT

First Ground of Rejection:

Claims 1, 9, 12, 20, and 23 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Johnson in view of Black. Appellant traverses this rejection for at least the following reasons.

Independent claims (by number):

Appellant respectfully submits that each of claims 1, 12, and 23 recites a combination of features not taught or suggested in Johnson or Black. For example, claim 1 recites a combination of features including: "wherein said plurality of functional units is grouped into a plurality of distinct issue slots, and wherein **each** of said plurality of **issue slots includes at least two functional units of different types**" and "wherein said arbitration logic is configured to **allocate portions of said read ports to functional units included in each of said plurality of issue slots**, and wherein said **portions are respectively** associated with said issue slots."

In the final Office action dated July 28, 2006, the Examiner acknowledges Johnson does not teach "wherein said plurality of **functional units is grouped into a plurality of distinct issue slots**, and wherein **each** of said plurality **of issue slots**

includes at least two functional units of different types" or "wherein said arbitration logic is configured to allocate **portions of said read ports to functional units included in each of said plurality of issue slots**, and wherein said **portions are respectively associated with said issue slots**." (Emphasis added)

However, the Examiner asserts that Black teaches these limitations. Specifically, the Examiner asserts "Black discloses a plurality of functional units being grouped into a plurality of distinct issue slots (paragraph 34)." Appellant respectfully disagrees with the Examiner's assertion and the Examiner's application of Black to Appellant's claims.

More particularly, Black discloses at paragraph [0034]

> "FIG. 8 illustrates one embodiment of the invention. The embodiment of the invention illustrated in FIG. 8 comprises distributed slow 801 and fast 805 windows, each corresponding to **a different execution cluster** 810. The distributed slow and fast windows allow latency-tolerant and latency-intolerant, respectively, **instructions to be scheduled and executed according to their latency characteristics rather than scheduling all of the latency-tolerant and latency-intolerant instructions f**or execution **by one slow execution cluster and by one fast execution cluster, respectively**." (Emphasis added)

Black also discloses in paragraph [0021]

> "Furthermore, **the execution unit may be part of an execution cluster in order to process instructions of a similar type or similar attributes, such as latency-toleranc**e. In other embodiments, the execution unit may be a single execution unit." (Emphasis added)

From the foregoing, Appellant submits Black does not disclose any aspect of grouping functional units into issue slots, where each issue slot includes different types of functional units, and performing port allocation on the basis of those issue slots. Instead, Black discloses slow and fast windows may be used for scheduling latency tolerant and latency-intolerant instructions to execution clusters having similar attributes (e.g., latency tolerance). Nothing about Black's technique suggests any aspect of grouping dissimilar functional units into multiple distinct issue slots and allocating portions of read ports in association with those issue slots. To the contrary, Black suggests a given execution cluster may process instructions of similar type or attributes, and that each window may correspond to a given execution cluster. Appellant submits Black is teaching grouping execution units into clusters based upon the instruction type and/or attribute. One skilled in the art would recognize that Black is merely one way of scheduling instructions to achieve instruction level parallelism (ILP), and it is orthogonal to grouping functional units based on issue slot. For example, if one were to recognize that the system most frequently encounters instruction streams in which many similar instructions may be issued, the system taught by Black may be advantageous, as having clusters that execute similar instructions may decrease latency. Moreover, the fact that the clusters may execute instructions with similar attributes teaches away from Appellant's invention.

In the Advisory action of November 21, 2006, the Examiner asserts "An "issue slot", traditionally, is interpreted in one of two ways. It is either the port of entry to a functional unit, or a position in memory that an instruction is stored before entering a functional unit." The Examiner also asserts "The fact that Black does not particularly say "issue slots" does not change the fact that Black separates the functional units in such a way that they have distinct issue slots."

In response to the Examiner's assertion, Appellant respectfully disagrees with the Examiner's definition of an issue slot. Appellant submits an issue slot is "a **heterogeneous** grouping of functional units that forms the unit of allocation of a resource." In other words, an issue slot may be thought of as a scheduling opportunity during which certain resources may be allocated to execute one or more instructions. This is in contrast to the Examiner's allegation that an issue slot is a single hardware entity such as a port of a functional unit or a memory position. The Appellant asserted in the response that Black does not teach "grouping functional units into issue slots, where each issue slot includes different types of functional units," and "allocating portions of read ports in association with those issue slots."

In addition, similar to other allegations, the Examiner's alleged meaning of an issue slot appears to contradict the meaning of the phrase as it used in the art. Appellant has on two different occasions requested the Examiner provide evidentiary support using either an affidavit or a reference that teaches what the Examiner alleges. Specifically, Appellant notes that for numerous ones of the rejected claims, the Examiner has acknowledged that Johnson fails to disclose the limitations of the claim, but has asserted that the limitations are "well-known in the art." Appellant traverses each such statement made by the Examiner. Appellant notes that the claimed limitations are directed to the invention as a whole, and asserts that applying the above-mentioned limitations to the claimed invention as a whole is not well known. Merely stating that individual aspects of a claimed invention are well known does not render the combination well known without some objective reason to combine the individual teachings. *Ex parte Levengood*, 28 USPQ2d 1300.

As the Court of Appeals for the Federal Circuit explained in *In re Sang Su Lee*, Docket No. 00-1158 (Fed. Cir. January 18, 2002), conclusory statements such as those provided by the Examiner that a claim limitation is well known or common knowledge do not fulfill the Examiner's obligation. "Deficiencies of the cited references cannot be remedied by the [Examiner's] general conclusions about what is 'basic knowledge' or 'common sense.'" *In re Zurko*, 59 USPQ2d 1693, 1697 (Fed. Cir. 2001). "Common knowledge and common sense ... do not substitute for authority." *In re Sang Su Lee*. Common knowledge "does not in and of itself make it so" absent evidence of such knowledge. *Smiths Industries Medical Systems, Inc. v. Vital Signs, Inc.*, 51 USPQ2d 1415, 1421 (Fed. Cir. 1999). Appellant therefore respectfully requested, in the Final Office action dated July 28, 2006, the Examiner provide either references that teach the claimed limitations of Appellants' application or an affidavit attesting to the Examiner's personal knowledge of same.

Appellant requested the same in the previous response dated May 8, 2006. The Examiner has continued to use this type of rejection rationale, as evidenced in the Advisory action dated November 21, 2006. Appellant notes that the use of terms like "extremely well-known in the art," although emphatic, do not relieve the Examiner of his obligation of either providing an affidavit or specific references. The Examiner still has not provided evidentiary support for many of his allegations.

For at least the above stated reasons, Appellant submits that the rejection of claims 1, 12, and 23 is in error and requests reversal of the rejection. The rejection of claim 9 (dependent from claim 1), and claim 20 (dependent from claim 12) are similarly in error for at least the above stated reasons, and reversal of the rejection is requested. Each of claims 9 and 20 recite additional combinations of features not taught or suggested in the cited art.

Second Ground of Rejection:

Claims 10-11 and 21-22 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Johnson in view of prior art. Appellant respectfully traverses this rejection and submits that the pending dependent claims are distinguishable for at least the reasons given above with respect to the independent claims.

Third Ground of Rejection:

Claims 24-37 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Johnson in view of Black. In addition to the reasons given above with respect to the independent claims, Appellant respectfully traverses this rejection for the following reasons.

Separately argued dependent claims (by number)

Claims 24 and 25, and 31 and 32 depend from claims 1 and 20 respectively. Accordingly, the rejection of claims 24 and 25, and 31 and 32 is in error for at least the reasons highlighted above with regard to claims 1. Additionally, each of claims 24 and 31 recite a combination of features including: "wherein said plurality of functional units includes arithmetic logic units and address generation units, and wherein each of said issue slots includes a respective one of said arithmetic logic units and a respective one of said address generation units."

Appellant submits these features are not taught or suggested in Johnson or Black as discussed above.

In the Final response dated July 28, 2006,the Examiner's alleges Johnson/Black teach " a processing system ultimately designed to reduce latency, would have been motivated to put and ALU and an address generation unit within the same issue slot if Johnson/Black wished to run programs that caused these two units to have a similar amount of latency... Appellant respectfully submits the latency of the ALU and AGU is in many cases transparent to program software execution. In addition, as described above Appellant submits Black is teaching grouping execution units into clusters based upon the instruction type (e.g., similar attributes and types). One skilled in the art would recognize that the system of Black is merely one way of scheduling instructions to achieve instruction level parallelism (ILP), and it is orthogonal to grouping functional units based on issue slot. For example, if one were to recognize that the system most frequently encounters instruction streams in which many similar instructions may be issued, the system taught by Black may be advantageous, as having clusters that execute similar instructions may decrease latency. However instruction streams that include many different types of instructions may not be appropriate for a system such as taught in Black. Accordingly, Appellant submits having each issue slot include an ALU and an AGU is not obvious in view of Johnson and Black.

Claims 25 and 32 recite features similar to "wherein said arbitration logic is further configured to allocate a given one of said portions of said read ports respectively associated with a given one of said plurality of issue slots to functional units included in said given issue slot in response to a scheduler scheduling operations to be executed in one or more functional units included in said given issue slot."

The Examiner asserts Black teaches this feature at paragraph 34 and 37. Appellant disagrees and further asserts Black teaches at paragraph 37

> "The register file is used to pass source data to instructions. Accordingly, the register file location may affect scheduling window capacity, size, and/or performance. In one embodiment of the invention the register file is located between a large scheduling window layer of the hierarchy and a smaller scheduling window layer of the hierarchy. In such an embodiment, the source data used by the registers need not be stored in the large scheduling window(s) along with the instruction and instead may be passed to the instruction after it is removed from the large scheduling window for scheduling or execution."

Appellant fails to see how this passage in Black teaches allocating read ports of the register file at all. In fact, this passage mentions the source data used by the registers… This would imply write ports of the register file, as the write port is the port used by the register file to get source data. In addition, as shown above, paragraph 34 of Black does not mention register file port allocation.

Accordingly, for at least the above stated reasons, Appellant submits that the rejection of claims 24 and 25, and 31 and 32 are in error and requests reversal of the rejection.

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejections of claims 1, 9-12, and 20-37 are erroneous, and reversal of the decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of $500.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5500-89200/SJC. This Appeal Brief is submitted with a return receipt postcard.

Respectfully submitted,

/ Stephen J. Curran /
Stephen J. Curran
Reg. No. 50,664
AGENT FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8843

Date: May 16, 2007

IX. CLAIMS APPENDIX

The claims on appeal are as follows.

1. A microprocessor comprising:

a plurality of functional units, wherein said plurality of functional units is configured to collectively operate on a maximum number of operands in a given execution cycle, wherein said plurality of functional units is grouped into a plurality of distinct issue slots, and wherein each of said plurality of issue slots includes at least two functional units of different types;

a register file configured to provide a number of read ports that is insufficient to provide said maximum number of operands to said plurality of functional units in said given execution cycle; and

an arbitration logic coupled to allocate said read ports of said register file for use by selected functional units during said given execution cycle, wherein said arbitration logic is configured to allocate portions of said read ports to functional units included in each of said plurality of issue slots, and wherein said portions are respectively associated with said issue slots.

9. The microprocessor as recited in claim 1 further comprising a scheduler configured to assign ones of a plurality of operations to respective ones of said plurality of functional units and to store an age of each of said plurality of operations, wherein said arbitration logic is further configured to allocate a first portion of said read ports to a

given one of said plurality of issue slots including a given respective functional unit dependent upon said age of a corresponding operation assigned to said given respective functional unit.

10. The microprocessor as recited in claim 1 further comprising a data forwarding unit configured to store results produced by said plurality of functional units, wherein said arbitration logic is configured to allocate said results to said selected functional units during said given execution cycle prior to allocating said read ports of said register file.

11. The microprocessor as recited in claim 1, wherein said arbitration logic is further configured to allocate a single read port of said register file to one or more of said selected functional units.

12. A method comprising:

configuring a plurality of functional units to collectively operate on a maximum number of operands in a given execution cycle, wherein said plurality of functional units is grouped into a plurality of distinct issue slots, and wherein each of said plurality of issue slots includes at least two functional units of different types;

providing a number of read ports that is insufficient to provide said maximum number of operands to said plurality of functional units in said given execution cycle; and allocating portions of said read ports of said register file to functional units included in each of said plurality of issue slots for use during said given execution cycle, wherein said portions are respectively associated with said issue slots.

20. The method as recited in claim 12 further comprising:

assigning ones of a plurality of operations to respective ones of said plurality of functional units;

storing an age of each of said plurality of operations; and

allocating a first portion of said read ports to a given one of said plurality of issue slots including a given respective functional unit dependent upon said age of a corresponding operation assigned to said given respective functional unit.

21. The method as recited in claim 12 further comprising:

storing results produced by said plurality of functional units in a data forwarding unit; and

allocating said results to said selected functional units during said given execution cycle prior to allocating said read ports of said register file.

22. The method as recited in claim 12 further comprising allocating a single read port of said register file to one or more of said selected functional units.

23. A system comprising:

a system memory including a plurality of addressable locations; and

a microprocessor coupled to said system memory and comprising:

a plurality of functional units, wherein said plurality of functional units is configured to collectively operate on a maximum number of operands in a given execution cycle, wherein said plurality of functional units is grouped into a plurality of distinct issue slots, and wherein each of said plurality of issue slots includes at least two functional units of different types;

a register file configured to provide a number of read ports that is insufficient to provide said maximum number of operands to said plurality of functional units in said given execution cycle; and

an arbitration logic coupled to allocate said read ports of said register file for use by selected functional units during said given execution cycle, wherein said arbitration logic is configured to allocate portions of said read ports to functional units included in each of said plurality of issue slots, and wherein said portions are respectively associated with said issue slots.

24. The microprocessor as recited in claim 1, wherein said plurality of functional units includes arithmetic logic units and address generation units, and wherein each of said issue slots includes a respective one of said arithmetic logic units and a respective one of said address generation units.

25. The microprocessor as recited in claim 1, wherein said arbitration logic is further configured to allocate a given one of said portions of said read ports respectively associated with a given one of said plurality of issue slots to functional units included in said given issue slot in response to a scheduler scheduling operations to be executed in one or more functional units included in said given issue slot.

26. The microprocessor as recited in claim 25 further comprising a data forwarding network configured to provide operands, wherein in response to said scheduler scheduling at least two operations to be respectively executed in at least two functional units included in said given issue slot during said given execution cycle, said arbitration logic is further configured to determine whether operand requirements of said at least two operations can be met by a combination of said given portion of said read ports and operands provided by said data forwarding network.

27. The microprocessor as recited in claim 26, wherein in response to determining that operand requirements of said at least two operations cannot be met by a combination of said given portion of said read ports and operands provided by said data forwarding network, said arbitration logic is further configured to direct said scheduler to reschedule at least one of said at least two operations and to allocate said given portion of said read ports to one or more other ones of said at least two operations.

28. The microprocessor as recited in claim 27, wherein said arbitration logic is further configured to allocate said given portion of said read ports to an oldest one of said at least two operations.

29. The microprocessor as recited in claim 1, wherein each of said portions of said read ports includes an equal number of said read ports.

30. The microprocessor as recited in claim 1, wherein at least two of said portions of said read ports include dissimilar numbers of said read ports.

31. The method as recited in claim 12, wherein said plurality of functional units includes arithmetic logic units and address generation units, and wherein each of said issue slots includes a respective one of said arithmetic logic units and a respective one of said address generation units.

32. The method as recited in claim 12, wherein said allocating said portions of said read ports further comprises allocating a given one of said portions of said read ports respectively associated with a given one of said plurality of issue slots to functional units included in said given issue slot in response to scheduling operations to be executed in one or more functional units included in said given issue slot.

33. The method as recited in claim 32, further comprising:

in response to at least two operations being scheduled to be respectively executed in at least two functional units included in said given issue slot during said given execution cycle, determining whether operand requirements of said at least two operations can be met by a combination of said given portion of said read ports and operands provided by a data forwarding network.

34. The method as recited in claim 33, further comprising:

in response to determining that operand requirements of said at least two operations cannot be met by a combination of said given portion of said read ports and operands provided by said data forwarding network, directing that at least one of said at least two operations be rescheduled and allocating said given portion of said read ports to one or more other ones of said at least two operations.

35. The method as recited in claim 34, further comprising allocating said given portion of said read ports to an oldest one of said at least two operations.

36. The method as recited in claim 12, wherein each of said portions of said read ports includes an equal number of said read ports.

37. The method as recited in claim 12, wherein at least two of said portions of said read ports include dissimilar numbers of said read ports.

X. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no related proceedings known to Appellant.